Carderock Division

Naval Surface Warfare Center

West Bethesda, Maryland 20817-5700

NSWCCD-50-TR-1998 / 012 December 1998

Hydromechanics Directorate

Research and Development Report

Uncertainty Analysis Applied to Least Squares Curve and Surface Fits

by

David E. Hess

William E. Smith

Approved for public release: distribution unlimited.

Preceding Page Blank

MAJOR CARDEROCK DIVISION TECHNICAL COMPONENTS

CODE	011	Director of Technology
	10	Machinery Systems/Programs and Logistics Directorate
	20	Ship Systems and Programs Directorate
	50	Hydromechanics Directorate
	60	Survivability, Structures and Materials Directorate
	70	Signatures Directorate
	80	Machinery Research and Development Directorate
	90	Machinery In-Service Engineering Directorate

CARDEROCK DIVISION, NSWC, ISSUES THREE TYPES OF REPORTS:

1. **CARDEROCKDIV reports, a formal series,** contain information of permanent technical value. They carry a consecutive numerical identification regardless of their classification or the originating directorate.

2. **Directorate reports, a semiformal series,** contain information of a preliminary, temporary, or proprietary nature or of limited interest or significance. They carry an alphanumerical identification issued by the originating directorate.

3. **Technical memoranda, an informal series,** contain technical documentation of limited use and interest. They are primarily working papers intended for internal use. They carry an identifying number which indicates their type and the numerical code of the originating directorate. Any distribution outside CARDEROCKDIV must be approved by the head of the originating directorate on a case-by-case basis.

NDW-DTRC 5602/50 (Rev. 9-94)

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY (Leave blank)	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	December 1998	Final

4. TITLE AND SUBTITLE
Uncertainty Analysis Applied to Least Squares Curve and Surface Fits

5. FUNDING NUMBERS
N0001497WX30267
PE 601153N
W/U 1-5030-327-20

6. AUTHOR(S)
David E. Hess and William E. Smith

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Carderock Division, Naval Surface Warfare Center
Code 5600
West Bethesda, Maryland 20817-5700

8. PERFORMING ORGANIZATION REPORT NUMBER
NSWCCD-50-TR-1998 / 012

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)
Dr. Patrick Purtell 703-696-4308
Office of Naval Research, Code333
800 N. Quincy St.
Arlington, Va. 22217-5660

Dr. Teresa McMullen
Office of Naval Research, Code 342

10. SPONSORING / MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION / AVAILABILITY STATEMENT
Approved for public release: distribution unlimited.

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*

The application of uncertainty analysis to curve and surface fits obtained by the method of Least Squares is described. The primary obstacle to the routine implementation of the uncertainty analysis procedure: the derivation of the sensitivity derivatives, has been removed. Analytic expressions for the derivatives applicable to fits of arbitrary order have been derived, and a step-by-step procedure for their incorporation within a computer program provided. A review of the techniques for the construction of curve and surface fits of arbitrary order is included. The use of uncertainty analysis as an aid to the assessment of the suitability of a particular fit is demonstrated by the application of these procedures to a series of examples employing a generic data set typical of experimentally derived data. Some results include: quantification of the penalty for using a higher-order fit when it is not appropriate, and the fact that reduction of uncertainty in the data to be fitted is more effective at reducing the uncertainty in fit coefficients and fitted values than simply increasing the amount of data used to construct the fit. A computer program implementing these procedures is available from the authors.

14. SUBJECT TERMS
Uncertainty analysis, least squares fits, numerical methods

15. NUMBER OF PAGES
41

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
Unclassified	Unclassified	Unclassified	

Contents

Figures

Tables

Abstract

The application of uncertainty analysis to curve and surface fits obtained by the method of Least Squares is described. The primary obstacle to the routine implementation of the uncertainty analysis procedure: the derivation of the sensitivity derivatives, has been removed. Analytic expressions for the derivatives applicable to fits of arbitrary order have been derived, and a step-by-step procedure for their incorporation within a computer program provided. A review of the techniques for the construction of curve and surface fits of arbitrary order is included. The use of uncertainty analysis as an aid to the assessment of the suitability of a particular fit is demonstrated by the application of these procedures to a series of examples employing a generic data set typical of experimentally derived data. Some results include: quantification of the penalty for using a higher-order fit when it is not appropriate, and the fact that reduction of uncertainty in the data to be fitted is more effective at reducing the uncertainty in fit coefficients and fitted values than simply increasing the amount of data used to construct the fit. A computer program implementing these procedures is available from the authors.

Administrative Information

This work was sponsored by the Office of Naval Research (Code 333) under Contract N0001497WX30267 and Program Element 601153N.

Introduction

The use of the method of Least Squares for the construction of curve fits as models for experimental behavior is, for better or worse, a universal practice. At an early stage of training in engineering disciplines and in the sciences, one is introduced to the procedure and quickly learns to produce linear and quadratic fits. Less well known is the general implementation of the procedure for polynomial fits of arbitrary order and the ability to replace the usual set of basis functions with alternatives which may be more appropriate. A useful example of the latter approach is the use of products of polynomials of arbitrary order for the construction of a surface fit to a function of two independent variables. The indiscriminate or incorrect use of the method can lead to faulty conclusions when modeling behavior, and because the technique is so popular and straightforward to apply, the potential for misuse is large.

Measures to assess the *goodness* of the fit exist and serve as a check on the suitability of a particular model, but these measures can be insensitive and, considered in isolation, misleading. One must also consider the degree to which measurement uncertainty existing in the data to be fitted will propagate into the fit coefficients and in fitted values obtained from the use of the model when judging the

appropriateness of a particular fit. This paper discusses the application of uncertainty analysis to curve and surface fits of arbitrary order determined using the method of Least Squares. The general procedure requires the calculation of various partial derivatives, termed *sensitivity derivatives*, which can be quite formidable if attempted in a brute-force fashion. Instead, analytic expressions for the computation of these derivatives when the method is applied to curve and surface fits have been found; these formulas may be implemented within a computer program, thereby making the use of uncertainty analysis an integral part of the procedure for the construction and assessment of an arbitrary order fit. The manner in which these derivatives are determined is linked to the procedure for calculating arbitrary order fits; for this reason, the general method for constructing curve and surface fits will be reviewed.

The calculation of uncertainty associated with coefficients for and with fitted values from a Least Squares fit is attributed to Coleman and Steele and may be found in the new edition of their text [1]. As of this writing, the text is scheduled to be released in February 1999. Because the method does not appear in their previous text and has been available only through private courses offered in the past few years, the uncertainty analysis procedure along with general instructions for its use have been reproduced here for convenience. Direct calculation of the sensitivity derivatives for the linear and quadratic curve fit cases is provided to allow an independent check on the implementation of the general method. A step-by-step outline for incorporating the general procedure within a computer program is included. All of the techniques described here have been executed in a FORTRAN computer program which is available upon request from the authors.

Finally, the use of uncertainty analysis as an aid to the assessment of the suitability of a particular fit is demonstrated by a series of simple examples employing a generic data set which is typical of experimentally derived data. The results illustrate the general use of the method, quantify certain previously held suspicions and introduce some surprises. The next section begins with a description of the general procedure for the derivation of an arbitrary order fit.

Computation of an Arbitrary Order Least Squares Curve Fit

Given n pairs of experimental data (x_i, y_i) for $i = 1, 2, \ldots, n$, one can fit an m^{th} order polynomial to the data of the form:

$$y = c_1 + c_2 x + c_3 x^2 + \ldots + c_{m+1} x^m = \sum_{k=0}^{m} c_{k+1} x^k \ , \tag{1}$$

where the choice of m is constrained such that $n \geq m+1$. The values of the unknown coefficients are determined by the criterion used to choose the best fit to the data. The Least Squares criterion chooses the coefficients of the fit such that the sum of the squares of the distances of the measured data (x_i, y_i) from the fitted data $(x_i, \hat{y}_i)$ is a minimum. In other words, we seek to minimize the quantity

$$q = \sum_{i=1}^{n} \left(y_i - \hat{y}_i\right)^2 = \sum_{i=1}^{n} \left(y_i - c_1 - c_2 x_i - c_3 x_i^2 - \ldots - c_{m+1} x_i^m\right)^2 \ . \tag{2}$$

Because the (x_i, y_i) are known, q will depend only upon the coefficients c_k for $k = 1, 2, \ldots, m+1$. To minimize q, we compute partial derivatives with respect to each coefficient and set each resulting equation equal to zero to obtain

$$
\begin{aligned}
\frac{\partial q}{\partial c_1} &= 2\sum_{i=1}^{n}\left(y_i - c_1 - c_2 x_i - c_3 x_i^2 - \ldots - c_{m+1} x_i^m\right)(-1) = 0 \\
\frac{\partial q}{\partial c_2} &= 2\sum_{i=1}^{n}\left(y_i - c_1 - c_2 x_i - c_3 x_i^2 - \ldots - c_{m+1} x_i^m\right)(-x_i) = 0 \\
\frac{\partial q}{\partial c_3} &= 2\sum_{i=1}^{n}\left(y_i - c_1 - c_2 x_i - c_3 x_i^2 - \ldots - c_{m+1} x_i^m\right)(-x_i^2) = 0 \; . \\
&\vdots \\
\frac{\partial q}{\partial c_{m+1}} &= 2\sum_{i=1}^{n}\left(y_i - c_1 - c_2 x_i - c_3 x_i^2 - \ldots - c_{m+1} x_i^m\right)(-x_i^m) = 0
\end{aligned}
\tag{3}
$$

Rearranging yields the following set of $m+1$ simultaneous equations:

$$
\begin{aligned}
c_1 n + c_2 \sum x_i + c_3 \sum x_i^2 + \ldots + c_{m+1} \sum x_i^m &= \sum y_i \\
c_1 \sum x_i + c_2 \sum x_i^2 + c_3 \sum x_i^3 + \ldots + c_{m+1} \sum x_i^{m+1} &= \sum x_i y_i \\
c_1 \sum x_i^2 + c_2 \sum x_i^3 + c_3 \sum x_i^4 + \ldots + c_{m+1} \sum x_i^{m+2} &= \sum x_i^2 y_i \quad . \\
&\vdots \\
c_1 \sum x_i^m + c_2 \sum x_i^{m+1} + c_3 \sum x_i^{m+2} + \ldots + c_{m+1} \sum x_i^{2m} &= \sum x_i^m y_i
\end{aligned}
\tag{4}
$$

These equations can be easily represented in matrix form and are known as the *normal equations*.

$$
\begin{bmatrix}
\sum x_i^0 & \sum x_i^1 & \cdots & \sum x_i^m \\
\sum x_i^1 & \sum x_i^2 & \cdots & \sum x_i^{m+1} \\
\vdots & \vdots & \vdots & \vdots \\
\sum x_i^m & \sum x_i^{m+1} & \cdots & \sum x_i^{2m}
\end{bmatrix}
\begin{bmatrix} c_1 \\ c_2 \\ \vdots \\ c_{m+1} \end{bmatrix}
=
\begin{bmatrix} \sum x_i^0 y_i \\ \sum x_i^1 y_i \\ \vdots \\ \sum x_i^m y_i \end{bmatrix} .
\tag{5}
$$

Here we have a $(m+1)\times(m+1)$ matrix of known quantities multiplying an unknown $(m+1)\times 1$ coefficient vector to obtain a known $(m+1)\times 1$ right-hand-side vector. There are a variety of methods available for the solution of a simultaneous system of linear equations. Note that the $(m+1)\times(m+1)$ matrix above is symmetric and positive-definite which implies that very efficient and fast simultaneous equation solvers may be employed for the solution. Alternatively, if the matrix is nearly singular, then singular value decomposition techniques may be applied. These procedures are described in Numerical Recipes[3].

To build the above set of equations within a computer code, one begins by forming the $n\times(m+1)$ matrix $\mathbf{C}$ and its $(m+1)\times n$ transpose $\mathbf{C}^{\mathbf{T}}$:

$$\mathbf{C} = \begin{bmatrix} x_1^0 & x_1^1 & \cdots & x_1^m \\ x_2^0 & x_2^1 & \cdots & x_2^m \\ \vdots & \vdots & \vdots & \vdots \\ x_n^0 & x_n^1 & \cdots & x_n^m \end{bmatrix} \quad \text{and} \quad \mathbf{C}^{\mathrm{T}} = \begin{bmatrix} x_1^0 & x_2^0 & \cdots & x_n^0 \\ x_1^1 & x_2^1 & \cdots & x_n^1 \\ \vdots & \vdots & \vdots & \vdots \\ x_1^m & x_2^m & \cdots & x_n^m \end{bmatrix} . \tag{6}$$

Then, the *normal equations* defined in Eqs. 5 can be easily developed using matrix multiplication as follows:

$$\mathbf{C}^{\mathrm{T}}\mathbf{C}\ \mathbf{c} = \mathbf{C}^{\mathrm{T}}\mathbf{y}\ , \tag{7}$$

and the known matrices $\mathbf{C}^{\mathrm{T}}\mathbf{C}$ and $\mathbf{C}^{\mathrm{T}}\mathbf{y}$ can be passed to a simultaneous equations solver to solve for the unknown coefficient vector $\mathbf{c}$.

Following Holman [4], quantities which describe the degree of *goodness* of the fit are the standard error of the fitted values given by

$$\begin{aligned} \sigma_{y,x} &= \left[\frac{1}{n-(m+1)} \sum_{i=1}^{n} \left(y_i - \hat{y}_i \right)^2 \right]^{1/2} \\ \sigma_{y,x} &= \left[\frac{1}{n-(m+1)} \sum_{i=1}^{n} \left(y_i - c_1 - c_2 x_i - c_3 x_i^{\,2} - \ldots - c_{m+1} x_i^{\,m} \right)^2 \right]^{1/2} , \end{aligned} \tag{8}$$

and the correlation coefficient, R, which is computed from

$$R = \left[1 - \frac{\sigma_{y,x}^2}{\sigma_y^2} \right]^{1/2} \quad \text{where the standard deviation of } y, \quad \sigma_y = \left[\frac{1}{n-1} \sum_{i=1}^{n} \left(y_i - \bar{y} \right)^2 \right]^{1/2} . \tag{9}$$

Note that some texts omit the $\dfrac{1}{n-\ldots}$ coefficients in the numerator and denominator of the expression that defines R. The coefficient of determination, used by some texts, is also used as a measure of the *goodness* of the fit and is simply the square of the correlation coefficient. This quantity is usually defined such that the $\dfrac{1}{n-\ldots}$ coefficients in the numerator and denominator are omitted.

Uncertainty Method

Given n pairs of experimental data $\left(x_i, y_i\right)$, along with associated bias errors, $\left(B_x\right)_i$ and $\left(B_y\right)_i$, and precision errors, $\left(P_x\right)_i$ and $\left(P_y\right)_i$, the method due to Coleman and Steele [1] determines how these errors propagate through the governing equations to produce uncertainty in the resulting fit coefficients and in future values predicted from the fit. Consider the simple case of a linear fit

$$y = c_1 + c_2 x\ . \tag{10}$$

The equations for the coefficients are the solutions to Eqs. 5 for the case $m = 1$ and are given by

$$c_1 = \frac{\sum_{i=1}^{n} x_i^2 \sum_{i=1}^{n} y_i - \sum_{i=1}^{n} x_i \sum_{i=1}^{n} x_i y_i}{n\sum_{i=1}^{n} x_i^2 - \left(\sum_{i=1}^{n} x_i\right)^2} \quad \text{and} \quad c_2 = \frac{n\sum_{i=1}^{n} x_i y_i - \sum_{i=1}^{n} x_i \sum_{i=1}^{n} y_i}{n\sum_{i=1}^{n} x_i^2 - \left(\sum_{i=1}^{n} x_i\right)^2} . \tag{11}$$

To compute uncertainty in fit coefficients, the method treats equations such as Eqs. 11 (or Eqs. 21 below) as propagation equations and then applies standard techniques for the propagation of uncertainty. These equations are presented in the next section. Then, combining Eqs. 10 and 11 one can track the uncertainty propagated into future fitted values which will be described in the following section.

Uncertainty in Curve Fit Coefficients

Denote the bias uncertainty propagated into the k^{th} coefficient c_k as $\left(B_c\right)_k$; this quantity may be computed from

$$\begin{aligned}\left(B_c\right)_k^2 &= \sum_{i=1}^{n}\left(\frac{\partial c_k}{\partial x_i}\right)^2 \left(B_x\right)_i^2 + \sum_{i=1}^{n}\left(\frac{\partial c_k}{\partial y_i}\right)^2 \left(B_y\right)_i^2 + 2\sum_{i=1}^{n-1}\sum_{j=i+1}^{n}\left(\frac{\partial c_k}{\partial x_i}\right)\left(\frac{\partial c_k}{\partial x_j}\right)\left(B_x\right)_i\left(B_x\right)_j \\ &+ 2\sum_{i=1}^{n-1}\sum_{j=i+1}^{n}\left(\frac{\partial c_k}{\partial y_i}\right)\left(\frac{\partial c_k}{\partial y_j}\right)\left(B_y\right)_i\left(B_y\right)_j + 2\sum_{i=1}^{n-1}\sum_{j=i+1}^{n}\left(\frac{\partial c_k}{\partial x_i}\right)\left(\frac{\partial c_k}{\partial y_j}\right)\left(B_x\right)_i\left(B_y\right)_j\end{aligned} , \tag{12}$$

where we have assumed correlated biases among the x_i, among the y_i and also between x_i and y_i. If any of these biases among the raw data values can be shown to be uncorrelated, then the corresponding terms in Eq. 12 should be excluded.

The precision uncertainty propagated into the coefficient c_k, $\left(P_c\right)_k$, is obtained from a similar equation; however, since precision errors are considered to be random, all precision errors among the raw data values are uncorrelated.

$$\left(P_c\right)_k^2 = \sum_{i=1}^{n}\left(\frac{\partial c_k}{\partial x_i}\right)^2 \left(P_x\right)_i^2 + \sum_{i=1}^{n}\left(\frac{\partial c_k}{\partial y_i}\right)^2 \left(P_y\right)_i^2 . \tag{13}$$

To arrive at an overall uncertainty, $\left(U_c\right)_k$, for the coefficient c_k, one must combine the bias uncertainty computed in Eq. 12 with the precision uncertainty obtained in Eq. 13. There are two generally accepted methods for computing $\left(U_c\right)_k$. The first method is denoted as the root-sum-square (RSS) method and the total uncertainty is given by

$$\left(U_c\right)_k = \sqrt{\left(B_c\right)_k^2 + \left(P_c\right)_k^2} . \tag{14}$$

This uncertainty is considered to be a 95% coverage estimate when B and P are 95% confidence values.

The second method is to combine bias and precision uncertainties by simply adding them (ADD method):

$$\left(U_c\right)_k = \left(B_c\right)_k + \left(P_c\right)_k \; . \tag{15}$$

This method produces a total uncertainty which provides about 99% coverage when B and P are 95% confidence values and when neither B nor P is negligible relative to the other. However, when either B or P is negligible relative to the other, clearly the total uncertainty cannot be better than a 95% confidence estimate. To carry out these computations, we need expressions for the partial derivatives (sensitivity derivatives) that appear in Eqs. 12 and 13. For the simple case of a linear fit, the derivatives are applied to Eqs. 11. We make use of the formula for the derivative of a quotient, and for the linear case, we have:

$$\frac{\partial c_k}{\partial x_i} = \frac{den \dfrac{\partial\, num_k}{\partial x_i} - num_k \dfrac{\partial\, den}{\partial x_i}}{den^2} = \frac{1}{den}\left[\frac{\partial\, num_k}{\partial x_i} - c_k \frac{\partial\, den}{\partial x_i}\right] , \tag{16}$$

$$\frac{\partial c_k}{\partial y_i} = \frac{den \dfrac{\partial\, num_k}{\partial y_i} - num_k \dfrac{\partial\, den}{\partial y_i}}{den^2} = \frac{1}{den}\left[\frac{\partial\, num_k}{\partial y_i} - c_k \frac{\partial\, den}{\partial y_i}\right] , \tag{17}$$

$$\frac{\partial\, num_1}{\partial x_i} = 2x_i \sum y_i - \sum x_i y_i - y_i \sum x_i \quad \text{and} \quad \frac{\partial\, num_1}{\partial y_i} = \sum x_i^2 - x_i \sum x_i \; , \tag{18}$$

$$\frac{\partial\, num_2}{\partial x_i} = n y_i - \sum y_i \quad \text{and} \quad \frac{\partial\, num_2}{\partial y_i} = n x_i - \sum x_i \; . \tag{19}$$

$$\frac{\partial\, den}{\partial x_i} = 2n x_i - 2\sum x_i \quad \text{and} \quad \frac{\partial\, den}{\partial y_i} = 0 \; . \tag{20}$$

For the case of a quadratic fit, the equations for the coefficients are the solutions to Eqs. 5 for the case $m = 2$ and are given by:

$$\begin{aligned}
c_1 &= \frac{\sum x_i^2 y_i\left[\sum x_i \sum x_i^3 - \left(\sum x_i^2\right)^2\right] + \sum x_i^3\left[\sum x_i^2 \sum x_i y_i - \sum x_i^3 \sum y_i\right] + \sum x_i^4\left[\sum x_i^2 \sum y_i - \sum x_i \sum x_i y_i\right]}{\sum x_i^2\left[\sum x_i \sum x_i^3 - \left(\sum x_i^2\right)^2\right] + \sum x_i^3\left[\sum x_i \sum x_i^2 - n\sum x_i^3\right] + \sum x_i^4\left[n\sum x_i^2 - \left(\sum x_i\right)^2\right]} \\
c_2 &= \frac{\sum x_i^2\left[\sum x_i^3 \sum y_i - \sum x_i^2 \sum x_i y_i\right] + \sum x_i^2 y_i\left[\sum x_i \sum x_i^2 - n\sum x_i^3\right] + \sum x_i^4\left[n\sum x_i y_i - \sum x_i \sum y_i\right]}{\sum x_i^2\left[\sum x_i \sum x_i^3 - \left(\sum x_i^2\right)^2\right] + \sum x_i^3\left[\sum x_i \sum x_i^2 - n\sum x_i^3\right] + \sum x_i^4\left[n\sum x_i^2 - \left(\sum x_i\right)^2\right]} \\
c_3 &= \frac{\sum x_i^2\left[\sum x_i \sum x_i y_i - \sum x_i^2 \sum y_i\right] + \sum x_i^3\left[\sum x_i \sum y_i - n\sum x_i y_i\right] + \sum x_i^2 y_i\left[n\sum x_i^2 - \left(\sum x_i^2\right)^2\right]}{\sum x_i^2\left[\sum x_i \sum x_i^3 - \left(\sum x_i^2\right)^2\right] + \sum x_i^3\left[\sum x_i \sum x_i^2 - n\sum x_i^3\right] + \sum x_i^4\left[n\sum x_i^2 - \left(\sum x_i\right)^2\right]}
\end{aligned} \tag{21}$$

Applying derivatives to these expressions is a tedious task. After much algebra, we find:

$$\begin{aligned}\frac{\partial\, num_1}{\partial\, x_i} &= \sum x_i\left[3x_i^2\sum x_i^2 y_i - 4x_i^3\sum x_i y_i\right] \\ &+\sum x_i^2\left[3x_i^2\sum x_i y_i + 4x_i^3\sum y_i - 2x_i y_i\sum x_i^2 - 4x_i\sum x_i^2 y_i\right] \\ &+\sum x_i^3\left[y_i\sum x_i^2 + 2x_i\sum x_i y_i - 6x_i^2\sum y_i + 2x_i y_i\sum x_i + \sum x_i^2 y_i\right] \\ &+\sum x_i^4\left[2x_i\sum y_i - \sum x_i y_i - y_i\sum x_i\right]\end{aligned} \tag{22}$$

$$\frac{\partial\, num_1}{\partial\, y_i} = \sum x_i^2\left[-x_i^2\sum x_i^2\right] + \sum x_i^3\left[x_i^2\sum x_i + x_i\sum x_i^2 - \sum x_i^3\right] + \sum x_i^4\left[\sum x_i^2 - x_i\sum x_i\right] \tag{23}$$

$$\begin{aligned}\frac{\partial\, num_2}{\partial\, x_i} &= \sum x_i\left[2x_i\sum x_i^2 y_i - 4x_i^3\sum y_i\right] + \sum x_i^2\left[2x_i y_i\sum x_i + 3x_i^2\sum y_i - y_i\sum x_i^2\right] \\ &+\sum x_i^3\left[2x_i\sum y_i - 2nx_i y_i\right] + \sum x_i^4\left[ny_i - \sum y_i\right] + \sum x_i y_i\left[4nx_i^3 - 4x_i\sum x_i^2\right] \\ &+\sum x_i^2 y_i\left[\sum x_i^2 - 3nx_i^2\right]\end{aligned} \tag{24}$$

$$\frac{\partial\, num_2}{\partial\, y_i} = \sum x_i^2\left[x_i^2\sum x_i - x_i\sum x_i^2\right] + \sum x_i^3\left[\sum x_i^2 - nx_i^2\right] + \sum x_i^4\left[nx_i - \sum x_i\right] \tag{25}$$

$$\begin{aligned}\frac{\partial\, num_3}{\partial\, x_i} &= \sum x_i\left[3x_i^2\sum y_i - 2x_i y_i\sum x_i\right] + \sum x_i^2\left[y_i\sum x_i - 4x_i\sum y_i + 2nx_i y_i + \sum x_i y_i\right] \\ &+\sum x_i^3\left[\sum y_i - ny_i\right] + \sum x_i y_i\left[2x_i\sum x_i - 3nx_i^2\right] + \sum x_i^2 y_i\left[2nx_i - 2\sum x_i\right]\end{aligned} \tag{26}$$

$$\frac{\partial\, num_3}{\partial\, y_i} = \sum x_i\left[-x_i^2\sum x_i\right] + \sum x_i^2\left[x_i\sum x_i + nx_i^2 - \sum x_i^2\right] + \sum x_i^3\left[\sum x_i - nx_i\right] \tag{27}$$

$$\begin{aligned}\frac{\partial\, den}{\partial\, x_i} &= \sum x_i\left[6x_i^2\sum x_i^2 + 4x_i\sum x_i^3 - 4x_i^3\sum x_i\right] + \sum x_i^2\left[4nx_i^3 - 6x_i\sum x_i^2\right] \\ &+\sum x_i^3\left[2\sum x_i^2 - 6nx_i^2\right] + \sum x_i^4\left[2nx_i - 2\sum x_i\right]\end{aligned} \tag{28}$$

$$\frac{\partial\, den}{\partial\, y_i} = 0 \tag{29}$$

These equations are then substituted into Eqs. 16 and 17 to find the needed derivatives. Summarizing, one computes the bias and precision uncertainty propagated into the fit coefficients from Eqs. 12 and 13 making use of Eqs. 11 and 16-20 for the $m = 1$ case or of Eqs. 16-17 and 21-29 for the $m = 2$ case. The total uncertainty in fit coefficients is then determined from Eqs. 14 or 15.

Uncertainty in Predicted Values from Curve Fit

After calculating the coefficients of the fit equation, one can supply a new value X in order to compute a new value Y using:

$$Y(X) = c_1 + c_2 X + c_3 X^2 + \ldots + c_{m+1} X^m \quad \text{for } X \text{ chosen such that } x_{\min} \le X \le x_{\max}\,. \tag{30}$$

We wish to consider the uncertainty that will propagate into this fitted *Y(X)*. Different equations are used depending upon whether or not there is uncertainty associated with X. The most general case is the one in which there *is* uncertainty associated with X. The bias uncertainty that propagates into the fitted *Y(X)* for this case may be computed from

$$\begin{aligned} B_Y^2 = {} & \sum_{i=1}^{n}\left(\frac{\partial Y}{\partial x_i}\right)^2 (B_x)_i^2 + \sum_{i=1}^{n}\left(\frac{\partial Y}{\partial y_i}\right)^2 (B_y)_i^2 + 2\sum_{i=1}^{n-1}\sum_{j=i+1}^{n}\left(\frac{\partial Y}{\partial x_i}\right)\left(\frac{\partial Y}{\partial x_j}\right)(B_x)_i (B_x)_j \\ & + 2\sum_{i=1}^{n-1}\sum_{j=i+1}^{n}\left(\frac{\partial Y}{\partial x_i}\right)\left(\frac{\partial Y}{\partial y_j}\right)(B_x)_i (B_y)_j + 2\sum_{i=1}^{n-1}\sum_{j=i+1}^{n}\left(\frac{\partial Y}{\partial y_i}\right)\left(\frac{\partial Y}{\partial y_j}\right)(B_y)_i (B_y)_j\,, \\ & + \left(\frac{\partial Y}{\partial X}\right)^2 B_X^2 + 2\sum_{i=1}^{n}\left(\frac{\partial Y}{\partial X}\right)\left(\frac{\partial Y}{\partial x_i}\right) B_X (B_x)_i + 2\sum_{i=1}^{n}\left(\frac{\partial Y}{\partial X}\right)\left(\frac{\partial Y}{\partial y_i}\right) B_X (B_y)_i \end{aligned} \tag{31}$$

where we have assumed the most general case of correlated biases among the x_i (and X), among the y_i (and X) and also between x_i and y_i. If, as described above, any of these biases can be shown to be uncorrelated then, of course, the appropriate terms in Eq. 31 should be excluded.

The precision uncertainty that propagates into the fitted *Y(X)* is obtained from

$$P_Y^2 = \sum_{i=1}^{n}\left(\frac{\partial Y}{\partial x_i}\right)^2 (P_x)_i^2 + \sum_{i=1}^{n}\left(\frac{\partial Y}{\partial y_i}\right)^2 (P_y)_i^2 + \left(\frac{\partial Y}{\partial X}\right)^2 P_X^2\,. \tag{32}$$

Now, to evaluate these expressions, we need to determine the partial derivatives: $\dfrac{\partial Y}{\partial x_i}$, $\dfrac{\partial Y}{\partial y_i}$ and $\dfrac{\partial Y}{\partial X}$.

One applies the derivatives to Eq. 30 which, for the linear case, may be written as

$$Y(X) = \left[\frac{n\sum\limits_{i=1}^{n} x_i y_i - \sum\limits_{i-1}^{n} x_i \sum\limits_{i-1}^{n} y_i}{n\sum\limits_{i=1}^{n} x_i^2 - \left(\sum\limits_{i-1}^{n} x_i\right)^2}\right] X + \left[\frac{\sum\limits_{i=1}^{n} x_i^2 \sum\limits_{i-1}^{n} y_i - \sum\limits_{i-1}^{n} x_i \sum\limits_{i=1}^{n} x_i y_i}{n\sum\limits_{i=1}^{n} x_i^2 - \left(\sum\limits_{i-1}^{n} x_i\right)^2}\right]. \tag{33}$$

The needed derivatives are found to be:

$$\frac{\partial Y}{\partial x_i} = \frac{\partial c_1}{\partial x_i} + \frac{\partial c_2}{\partial x_i} X + \frac{\partial c_3}{\partial x_i} X^2 + \ldots + \frac{\partial c_{m+1}}{\partial x_i} X^m \ , \tag{34}$$

$$\frac{\partial Y}{\partial y_i} = \frac{\partial c_1}{\partial y_i} + \frac{\partial c_2}{\partial y_i} X + \frac{\partial c_3}{\partial y_i} X^2 + \ldots + \frac{\partial c_{m+1}}{\partial y_i} X^m \ , \text{ and} \tag{35}$$

$$\frac{\partial Y}{\partial X} = c_2 + 2c_3 X + 3c_4 X^2 + \ldots + mc_{m+1} X^{m-1} \ , \tag{36}$$

where $\frac{\partial c_k}{\partial x_i}$ and $\frac{\partial c_k}{\partial y_i}$ are as calculated previously for the linear or quadratic cases above. Note that $\frac{\partial Y}{\partial x_i}$, $\frac{\partial Y}{\partial y_i}$ and $\frac{\partial Y}{\partial X}$ are functions of X and must be recomputed for each new value of X.

Now, given a value X, we can compute a value Y from Eq. 30 and an associated B_Y and P_Y from Eqs. 31 and 32. From the latter two quantities we can determine a total uncertainty U_Y from Eqs. 14 or 15. This process should be repeated for X_j which vary throughout the range $x_{\min} \le X_j \le x_{\max}$, and a series of $Y(X)_j$ and $(U_Y)_j$ will result. Then, the uncertainty in fitted $Y(X)_j$ values should be plotted along with the fitted curve as follows. Plot the original (x_i, y_i) data values along with the fitted curve $Y(X)_j = c_1 + c_2 X_j + c_3 X_j^2 + \ldots + c_{m+1} X_j^m$ obtained from the new (X_j, Y_j) data pairs. Then plot above and below this fitted curve the two additional curves: $Y(X)_j + (U_Y)_j$ and $Y(X)_j - (U_Y)_j$. The latter two curves then give an indication of the total uncertainty in fitted $Y(X)_j$ values across the entire range $x_{\min} \le X_j \le x_{\max}$.

Summary of Fitting Procedure

To summarize, then, one can compute the total uncertainty in the fit coefficients as well as in the fitted Y_i values by performing the following steps:

1. From the (x_i, y_i) data pairs, compute the fit coefficients. Examples for the linear and quadratic cases are given in Eqs. 11 and 21, respectively.

2. From the (x_i, y_i) data pairs, compute the sensitivity derivatives. Examples for the linear and quadratic cases are given in Eqs. 16-20 and Eqs. 16-17, 22-29, respectively.

3. Compute uncertainty in fit coefficients $(B_c)_k$ and $(P_c)_k$ from Eqs. 12 and 13, respectively.

4. Form the total uncertainty $(U_c)_k$ from Eqs. 14 or 15.

5. Supply a value X_j in the range $x_{\min} \leq X_j \leq x_{\max}$, and for this X_j compute $Y(X)_j$ from Eq. 30 and $\frac{\partial Y}{\partial x_i}$, $\frac{\partial Y}{\partial y_i}$ and $\frac{\partial Y}{\partial X}$ from Eqs. 34-36.

6. For this X_j, estimate B_X and P_X.

7. For this X_j, compute B_Y and P_Y from Eqs. 31 and 32, respectively.

8. For this X_j, form $(U_Y)_j$ from Eqs. 14 or 15.

9. Repeat steps 5 to 8 for X_j throughout the range $(x_i)_{min} \leq X_j \leq (x_i)_{max}$.

10. Plot the original (x_i, y_i) data values along with the fitted curve $Y(X)_j = c_1 + c_2 X_j + c_3 X_j^2 + \ldots + c_{m+1} X_j^m$ obtained from the new (X_j, Y_j) data pairs generated in step 9.

11. Then plot above and below, respectively, the curve generated in step 12, the two additional curves: $Y(X)_j + (U_Y)_j$ and $Y(X)_j - (U_Y)_j$. The latter two curves then give an indication of the total uncertainty in fitted $Y(X)_j$ values across the entire range $x_{\min} \leq X_j \leq x_{\max}$.

Calculation of Sensitivity Derivatives

As can be seen from the previous sections, the method is straightforward to apply; the primary difficulty is the derivation of the sensitivity derivatives for a fit of arbitrary order, and we will show how this may be accomplished in this section. One proceeds by first developing analytic expressions for the coefficients. This can be done by employing Cramer's rule for the solution of the simultaneous equations in Eqs. 5. Cramer's rule finds the solution for each coefficient in $\mathbf{c}$ as a quotient of determinants; the denominator is simply the determinant of the $\mathbf{C}^{\mathrm{T}}\mathbf{C}$ matrix for all coefficients, and the numerator is the determinant of the matrix formed by replacing the k^{th} column of the $\mathbf{C}^{\mathrm{T}}\mathbf{C}$ matrix with $\mathbf{C}^{\mathrm{T}}\mathbf{y}$ for coefficient c_k. Therefore, for a nonsingular $\mathbf{C}^{\mathrm{T}}\mathbf{C}$ matrix we can write:

$$c_1 = \frac{\begin{vmatrix} \sum x_i^0 y_i & \sum x_i^1 & \cdots & \sum x_i^m \\ \sum x_i^1 y_i & \sum x_i^2 & \cdots & \sum x_i^{m+1} \\ \vdots & \vdots & \vdots & \vdots \\ \sum x_i^m y_i & \sum x_i^{m+1} & \cdots & \sum x_i^{2m} \end{vmatrix}}{\begin{vmatrix} \sum x_i^0 & \sum x_i^1 & \cdots & \sum x_i^m \\ \sum x_i^1 & \sum x_i^2 & \cdots & \sum x_i^{m+1} \\ \vdots & \vdots & \vdots & \vdots \\ \sum x_i^m & \sum x_i^{m+1} & \cdots & \sum x_i^{2m} \end{vmatrix}} \quad \cdots \quad c_{m+1} = \frac{\begin{vmatrix} \sum x_i^0 & \sum x_i^1 & \cdots & \sum x_i^0 y_i \\ \sum x_i^1 & \sum x_i^2 & \cdots & \sum x_i^1 y_i \\ \vdots & \vdots & \vdots & \vdots \\ \sum x_i^m & \sum x_i^{m+1} & \cdots & \sum x_i^m y_i \end{vmatrix}}{\begin{vmatrix} \sum x_i^0 & \sum x_i^1 & \cdots & \sum x_i^m \\ \sum x_i^1 & \sum x_i^2 & \cdots & \sum x_i^{m+1} \\ \vdots & \vdots & \vdots & \vdots \\ \sum x_i^m & \sum x_i^{m+1} & \cdots & \sum x_i^{2m} \end{vmatrix}} . \tag{37}$$

The sensitivity derivatives that are needed for uncertainty calculations are the quantities: $\dfrac{\partial c_k}{\partial x_i}$ and $\dfrac{\partial c_k}{\partial y_i}$ for $i = 1, 2, \ldots, n$ and $k = 1, 2, \ldots, m+1$. As an example of these calculations, consider the computation of $\dfrac{\partial c_k}{\partial x_i}$.

$$\frac{\partial c_k}{\partial x_i} = \frac{\partial}{\partial x_i} \frac{\begin{vmatrix} \sum x_i^0 & \cdots & \sum x_i^0 y_i & \cdots & \sum x_i^m \\ \sum x_i^1 & \cdots & \sum x_i^1 y_i & \cdots & \sum x_i^{m+1} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \sum x_i^m & \cdots & \sum x_i^m y_i & \cdots & \sum x_i^{2m} \end{vmatrix}}{\begin{vmatrix} \sum x_i^0 & \sum x_i^1 & \cdots & \sum x_i^m \\ \sum x_i^1 & \sum x_i^2 & \cdots & \sum x_i^{m+1} \\ \vdots & \vdots & \vdots & \vdots \\ \sum x_i^m & \sum x_i^{m+1} & \cdots & \sum x_i^{2m} \end{vmatrix}} \tag{38}$$

$$= \frac{den \dfrac{\partial\, num}{\partial x_i} - num \dfrac{\partial\, den}{\partial x_i}}{den^2} = \frac{1}{den} \left[\frac{\partial\, num}{\partial x_i} - c_k \frac{\partial\, den}{\partial x_i} \right]$$

The computation reduces to the determination of derivatives of determinants in the numerator and the denominator. To see how this may be accomplished for any values of m and n, we will first consider the simple case of a 3×3 determinant.

$$
\begin{aligned}
\frac{\partial}{\partial x_i}\begin{vmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{vmatrix} &= \frac{\partial}{\partial x_i}\begin{bmatrix} +a_{11}a_{22}a_{33}+a_{21}a_{32}a_{13}+a_{31}a_{12}a_{23} \\ -a_{31}a_{22}a_{13}-a_{32}a_{23}a_{11}-a_{33}a_{21}a_{12} \end{bmatrix} \\
&= \begin{bmatrix} +\frac{\partial a_{11}}{\partial x_i}a_{22}a_{33}+\frac{\partial a_{21}}{\partial x_i}a_{32}a_{13}+\frac{\partial a_{31}}{\partial x_i}a_{12}a_{23} \\ -\frac{\partial a_{31}}{\partial x_i}a_{22}a_{13}-\frac{\partial a_{32}}{\partial x_i}a_{23}a_{11}-\frac{\partial a_{33}}{\partial x_i}a_{21}a_{12} \end{bmatrix} \\
&+ \begin{bmatrix} +a_{11}\frac{\partial a_{22}}{\partial x_i}a_{33}+a_{21}\frac{\partial a_{32}}{\partial x_i}a_{13}+a_{31}\frac{\partial a_{12}}{\partial x_i}a_{23} \\ -a_{31}\frac{\partial a_{22}}{\partial x_i}a_{13}-a_{32}\frac{\partial a_{23}}{\partial x_i}a_{11}-a_{33}\frac{\partial a_{21}}{\partial x_i}a_{12} \end{bmatrix} \\
&+ \begin{bmatrix} +a_{11}a_{22}\frac{\partial a_{33}}{\partial x_i}+a_{21}a_{32}\frac{\partial a_{13}}{\partial x_i}+a_{31}a_{12}\frac{\partial a_{23}}{\partial x_i} \\ -a_{31}a_{22}\frac{\partial a_{13}}{\partial x_i}-a_{32}a_{23}\frac{\partial a_{11}}{\partial x_i}-a_{33}a_{21}\frac{\partial a_{12}}{\partial x_i} \end{bmatrix} \\
&= \begin{vmatrix} \frac{\partial a_{11}}{\partial x_i} & a_{12} & a_{13} \\ \frac{\partial a_{21}}{\partial x_i} & a_{22} & a_{23} \\ \frac{\partial a_{31}}{\partial x_i} & a_{32} & a_{33} \end{vmatrix} + \begin{vmatrix} a_{11} & \frac{\partial a_{12}}{\partial x_i} & a_{13} \\ a_{21} & \frac{\partial a_{22}}{\partial x_i} & a_{23} \\ a_{31} & \frac{\partial a_{32}}{\partial x_i} & a_{33} \end{vmatrix} + \begin{vmatrix} a_{11} & a_{12} & \frac{\partial a_{13}}{\partial x_i} \\ a_{21} & a_{22} & \frac{\partial a_{23}}{\partial x_i} \\ a_{31} & a_{32} & \frac{\partial a_{33}}{\partial x_i} \end{vmatrix}.
\end{aligned} \tag{39}
$$

The evaluation of a derivative of the general case of an $(m+1)\times(m+1)$ determinant can be carried out by computing the sum of $m+1$ determinants; the k^{th} determinant in the sum is formed by replacing each element in the k^{th} column by its derivative with all other columns remaining the same. This is an important observation because each of the determinants in the sum can be formed efficiently within a computer code. For example, if one wished to compute the derivatives $\frac{\partial c_1}{\partial x_i}$ for the case $m=2$, one would form the quantities:

$$\frac{\partial c_1}{\partial x_i} = \frac{\partial}{\partial x_i} \frac{\begin{vmatrix} \sum x_i^0 y_i & \sum x_i^1 & \sum x_i^2 \\ \sum x_i^1 y_i & \sum x_i^2 & \sum x_i^3 \\ \sum x_i^2 y_i & \sum x_i^3 & \sum x_i^4 \end{vmatrix}}{\begin{vmatrix} \sum x_i^0 & \sum x_i^1 & \sum x_i^2 \\ \sum x_i^1 & \sum x_i^2 & \sum x_i^3 \\ \sum x_i^2 & \sum x_i^3 & \sum x_i^4 \end{vmatrix}} , \tag{40}$$

$$= \frac{den \dfrac{\partial num}{\partial x_i} - num \dfrac{\partial den}{\partial x_i}}{den^2} = \frac{1}{den}\left[\frac{\partial num}{\partial x_i} - c_1 \frac{\partial den}{\partial x_i}\right]$$

where the derivatives of the numerator and denominator would be found from

$$\frac{\partial num}{\partial x_i} = \begin{vmatrix} 0 & \sum x_i^1 & \sum x_i^2 \\ 1x_i^0 y_i & \sum x_i^2 & \sum x_i^3 \\ 2x_i^1 y_i & \sum x_i^3 & \sum x_i^4 \end{vmatrix} + \begin{vmatrix} \sum x_i^0 y_i & 1x_i^0 & \sum x_i^2 \\ \sum x_i^1 y_i & 2x_i^1 & \sum x_i^3 \\ \sum x_i^2 y_i & 3x_i^2 & \sum x_i^4 \end{vmatrix} + \begin{vmatrix} \sum x_i^0 y_i & \sum x_i^1 & 2x_i^1 \\ \sum x_i^1 y_i & \sum x_i^2 & 3x_i^2 \\ \sum x_i^2 y_i & \sum x_i^3 & 4x_i^3 \end{vmatrix} . \tag{41}$$

$$\frac{\partial den}{\partial x_i} = \begin{vmatrix} 0 & \sum x_i^1 & \sum x_i^2 \\ 1x_i^0 & \sum x_i^2 & \sum x_i^3 \\ 2x_i^1 & \sum x_i^3 & \sum x_i^4 \end{vmatrix} + \begin{vmatrix} \sum x_i^0 & 1x_i^0 & \sum x_i^2 \\ \sum x_i^1 & 2x_i^1 & \sum x_i^3 \\ \sum x_i^2 & 3x_i^2 & \sum x_i^4 \end{vmatrix} + \begin{vmatrix} \sum x_i^0 & \sum x_i^1 & 2x_i^1 \\ \sum x_i^1 & \sum x_i^2 & 3x_i^2 \\ \sum x_i^2 & \sum x_i^3 & 4x_i^3 \end{vmatrix}$$

To obtain the entire set of derivatives, $\dfrac{\partial c_1}{\partial x_i}$, for $i = 1, 2, \ldots, n$, one must form the quantities given in Eqs. 40 and 41 for each i for $i = 1, 2, \ldots, n$. Similarly, to compute $\dfrac{\partial c_1}{\partial y_i}$, for the case $m = 2$, one would form the quantities:

$$\frac{\partial c_1}{\partial y_i} = \frac{\partial}{\partial y_i} \frac{\begin{vmatrix} \sum x_i^0 y_i & \sum x_i^1 & \sum x_i^2 \\ \sum x_i^1 y_i & \sum x_i^2 & \sum x_i^3 \\ \sum x_i^2 y_i & \sum x_i^3 & \sum x_i^4 \end{vmatrix}}{\begin{vmatrix} \sum x_i^0 & \sum x_i^1 & \sum x_i^2 \\ \sum x_i^1 & \sum x_i^2 & \sum x_i^3 \\ \sum x_i^2 & \sum x_i^3 & \sum x_i^4 \end{vmatrix}} , \tag{42}$$

$$= \frac{den \dfrac{\partial num}{\partial y_i} - num \dfrac{\partial den}{\partial y_i}}{den^2} = \frac{1}{den}\left[\frac{\partial num}{\partial y_i} - c_1 \frac{\partial den}{\partial y_i}\right]$$

where the derivatives of the numerator and denominator would be found from

$$\frac{\partial\, num}{\partial y_i} = \begin{vmatrix} x_i^0 & \sum x_i^1 & \sum x_i^2 \\ x_i^1 & \sum x_i^2 & \sum x_i^3 \\ x_i^2 & \sum x_i^3 & \sum x_i^4 \end{vmatrix} + \begin{vmatrix} \sum x_i^0 y_i & 0 & \sum x_i^2 \\ \sum x_i^1 y_i & 0 & \sum x_i^3 \\ \sum x_i^2 y_i & 0 & \sum x_i^4 \end{vmatrix} + \begin{vmatrix} \sum x_i^0 y_i & \sum x_i^1 & 0 \\ \sum x_i^1 y_i & \sum x_i^2 & 0 \\ \sum x_i^2 y_i & \sum x_i^3 & 0 \end{vmatrix}$$
$$= \begin{vmatrix} x_i^0 & \sum x_i^1 & \sum x_i^2 \\ x_i^1 & \sum x_i^2 & \sum x_i^3 \\ x_i^2 & \sum x_i^3 & \sum x_i^4 \end{vmatrix} \qquad (43)$$
$$\frac{\partial\, den}{\partial y_i} = \begin{vmatrix} 0 & \sum x_i^1 & \sum x_i^2 \\ 0 & \sum x_i^2 & \sum x_i^3 \\ 0 & \sum x_i^3 & \sum x_i^4 \end{vmatrix} + \begin{vmatrix} \sum x_i^0 & 0 & \sum x_i^2 \\ \sum x_i^1 & 0 & \sum x_i^3 \\ \sum x_i^2 & 0 & \sum x_i^4 \end{vmatrix} + \begin{vmatrix} \sum x_i^0 & \sum x_i^1 & 0 \\ \sum x_i^1 & \sum x_i^2 & 0 \\ \sum x_i^2 & \sum x_i^3 & 0 \end{vmatrix} = 0$$

Here we see that the calculations are simpler requiring the calculation of only one determinant for the numerator (true, for any order *m*) and with the denominator not a function of y_i.

Summarizing, the required sensitivity derivatives, $\frac{\partial c_k}{\partial x_i}$ and $\frac{\partial c_k}{\partial y_i}$ for $i = 1, 2, \dots, n$ and $k = 1, 2, \dots, m+1$ consist of two $n \times (m+1)$ arrays. A recipe describing one possible implementation of these calculations within a computer code for any *m* and *n* is as follows.

1. For the computations dealing with coefficient c_k, form a set of $m+1$ arrays of size $(m+1)\times(m+1)$ and initially fill the arrays with the elements of the $\mathbf{C}^\mathrm{T}\mathbf{C}$ matrix. For coefficient c_k, replace the $j = k^{th}$ column of each of these $m+1$ arrays with $\mathbf{C}^\mathrm{T}\mathbf{y}$.
2. Repeat step 1 for each of the $m+1$ coefficients storing all data; these arrays will be used for numerator computations. Repeat step 1, one additional time ($m+2^{nd}$ time), where each of the arrays are filled with the elements of the $\mathbf{C}^\mathrm{T}\mathbf{C}$ matrix only. This latter set of arrays will be used for denominator computations. Note that all of this data can be stored within a four-dimensional array with the following dimensions: $m+1$ rows indexed by I (where I is a different index than *i* which indexes original data values), $m+1$ columns indexed by j, $m+1$ matrices required for derivative calculations for each coefficient indexed by *l*, and $m+2$ sets corresponding to $m+1$ numerator calculations for each coefficient and one denominator calculation (common to all coefficients) indexed by *k*. Denoting the four-dimensional array by A_{Ijlk} and looping over all four indices, we have

$$A_{Ijlk} = \begin{cases} CTY_I & j = k \\ CTC_{Ij} & j \neq k \end{cases}.$$

Note that the *k* index can loop to $m+2$ as required since *j* cannot equal *k* for this case.

3. Now, for the k^{th} set, replace the $j = l^{th}$ column in the l^{th} matrix in the set with $\dfrac{\partial}{\partial x_i}$ of the previous column contents. With loops over I, l and k this can be accomplished as follows.

$$A_{Ilk} = \begin{cases} \begin{cases} 0 & I = 1 \\ (I-1)x_i^{I-2}y_i & \text{otherwise} \end{cases} & l = k \\ \begin{cases} 0 & I + l < 3 \\ (I+l-2)x_i^{I+l-3} & \text{otherwise} \end{cases} & l \neq k \end{cases}$$

4. Repeat step 3 for each of the $m+2$ sets indexed by k.
5. Compute the determinant of each of the $m+1$ arrays per set. Sum these determinants and store the sum in a one dimensional array indexed by k.
6. Repeat step 5 for each of the $m+2$ sets. These sums represent $\dfrac{\partial\, num}{\partial x_i}$ for each of the $m+1$ coefficients with the last set corresponding to $\dfrac{\partial\, den}{\partial x_i}$. At this point all of the data in the four-dimensional array formed in step 2 has been used, and we proceed to the $\dfrac{\partial}{\partial y_i}$ calculations.
7. For the k^{th} set, replace the $j = l^{th}$ column in the l^{th} matrix in the set of $m+1$ arrays with $\dfrac{\partial}{\partial y_i}$ of the previous column contents. One may use the four-dimensional array as it exists after step 6; it does not have to be recreated. With loops over I, l and k this can be accomplished as follows.

$$A_{Ilk} = \begin{cases} 0 & \text{otherwise} \\ x_i^{I-1} & l = k \end{cases}.$$

8. Repeat step 7 for $m+1$ sets indexed by k. Do not bother with the $m+2^{nd}$ set because the denominator is not a function of y_i.
9. Compute only the determinant of the $l = k^{th}$ array in the set since the other determinants in the set will be zero. Store this determinant in a one-dimensional array indexed by k.

10. Repeat step 9 for $m+1$ sets and set the $m+2^{nd}$ element of the one-dimensional array equal to zero. These elements represent $\dfrac{\partial\, num}{\partial y_i}$ for each of the $m+1$ coefficients with the last element corresponding to $\dfrac{\partial\, den}{\partial y_i}=0$.

11. Using the derivatives computed in steps 6 and 10, compute the derivatives $\dfrac{\partial c_k}{\partial x_i}$ and $\dfrac{\partial c_k}{\partial y_i}$ for the k^{th} coefficient using Eqs. 40 and 42. Store the results in two two-dimensional arrays with dimensions $n \times (m+1)$ where the first dimension corresponds to $i = 1, 2, \ldots, n$ original data values and the second to $k = 1, 2, \ldots, m+1$ coefficients.

12. Finally, repeat steps 1-11 for $i = 1, 2, \ldots, n$ filling the arrays described in step 11.

To debug the implementation one can independently calculate the derivatives for the cases $m = 1$ and $m = 2$ by programming within a subroutine the derivative expressions derived above in an earlier section. Results obtained from this routine can then be compared to the results obtained from the brute-force subroutine.

Computation of an Arbitrary Order Least Squares Surface Fit

Given triplets of experimental data (x_i, y_i, z_i) for $i = 1, 2, \ldots, n$, where the data are presumed to be related by a function of the form $z = f(x, y)$, one can construct a surface fit with a functional form obtained from the product of two m^{th} order polynomials:

$$z = \left[\sum_{p=0}^{mx} a_{p+1}\, x^p\right]\left[\sum_{q=0}^{my} b_{q+1}\, y^q\right] = \sum_{p=0}^{mx}\sum_{q=0}^{my} c_k\, x^p y^q \text{ , where } k = q\,(m_x+1)+p+1 \text{ ,} \tag{44}$$

and there are $(m_x+1)(m_y+1)$ terms in the sum. Note that for a given amount of data, the choice of m_x and m_y is constrained by the condition that $n \geq (m_x+1)(m_y+1)$. When $m_x = m_y = 2$, we have:

$$z = c_1 + c_2 x + c_3 x^2 + c_4 y + c_5 xy + c_6 x^2 y + c_7 y^2 + c_8 xy^2 + c_9 x^2 y^2 \ . \tag{45}$$

An advantage to this formulation is that one can choose m_x and m_y to be different values. For example, if one believes the data to be linear in one independent variable and quadratic in the other, then a surface could be constructed using $m_x = 1$ and $m_y = 2$. For even finer control, an implementation of this method in a computer program could, for a given choice of m_x and m_y, allow the user to retain or omit any terms in the desired functional form. This is easy to accomplish and will be described further below.

The values of the unknown coefficients are determined by the Least Squares criterion. Therefore, we seek to minimize the quantity

$$q=\sum_{i=1}^{n}\left(z_i-\hat{z}_i\right)^2=\sum_{i=1}^{n}\left(z_i-c_1-c_2x_i-\ldots-c_ky_i-c_{k+1}x_iy_i-\ldots-c_{(mx+1)(my+1)}x_i^{mx}y_i^{my}\right)^2 . \tag{46}$$

Because the $\left(x_i, y_i, z_i\right)$ are known, q will depend only upon the coefficients c_k for $k=1,2,\ldots,(m_x+1)(m_y+1)$. To minimize q, we compute partial derivatives with respect to each coefficient and set each resulting equation equal to zero to obtain

$$\begin{aligned}
\frac{\partial q}{\partial c_1}&=2\sum_{i=1}^{n}\left(z_i-c_1-c_2x_i-\ldots-c_ky_i-c_{k+1}x_iy_i-\ldots-c_{(mx+1)(my+1)}x_i^{mx}y_i^{my}\right)(-1)=0\\
\frac{\partial q}{\partial c_2}&=2\sum_{i=1}^{n}\left(z_i-c_1-c_2x_i-\ldots-c_ky_i-c_{k+1}x_iy_i-\ldots-c_{(mx+1)(my+1)}x_i^{mx}y_i^{my}\right)\left(-x_i\right)=0\\
&\vdots\\
\frac{\partial q}{\partial c_k}&=2\sum_{i=1}^{n}\left(z_i-c_1-c_2x_i-\ldots-c_ky_i-c_{k+1}x_iy_i-\ldots-c_{(mx+1)(my+1)}x_i^{mx}y_i^{my}\right)\left(-y_i\right)=0\\
\frac{\partial q}{\partial c_{k+1}}&=2\sum_{i=1}^{n}\left(z_i-c_1-c_2x_i-\ldots-c_ky_i-c_{k+1}x_iy_i-\ldots-c_{(mx+1)(my+1)}x_i^{mx}y_i^{my}\right)\left(-x_iy_i\right)=0\\
&\vdots\\
\frac{\partial q}{\partial c_{(mx+1)(my+1)}}&=2\sum_{i=1}^{n}\left(z_i-c_1-c_2x_i-\ldots-c_ky_i-c_{k+1}x_iy_i-\ldots-c_{(mx+1)(my+1)}x_i^{mx}y_i^{my}\right)\left(-x_i^{mx}y_i^{my}\right)=0
\end{aligned} \tag{47}$$

Rearranging yields the following set of $(m_x+1)(m_y+1)$ simultaneous equations:

$$\begin{aligned}
c_1n+c_2\sum x_i+\ldots+c_k\sum y_i+c_{k+1}\sum x_iy_i+\ldots+c_{(mx+1)(my+1)}\sum x_i^{mx}y_i^{my}&=\sum z_i\\
c_1\sum x_i+c_2\sum x_i^2+\ldots+c_k\sum x_iy_i+c_{k+1}\sum x_i^2y_i+\ldots+c_{(mx+1)(my+1)}\sum x_i^{mx+1}y_i^{my}&=\sum x_iz_i\\
&\vdots\\
c_1\sum y_i+c_2\sum x_iy_i+\ldots+c_k\sum y_i^2+c_{k+1}\sum x_iy_i^2+\ldots+c_{(mx+1)(my+1)}\sum x_i^{mx}y_i^{my+1}&=\sum y_iz_i\\
c_1\sum x_iy_i+c_2\sum x_i^2y_i+\ldots+c_k\sum x_iy_i^2+c_{k+1}\sum x_i^2y_i^2+\ldots+c_{(mx+1)(my+1)}\sum x_i^{mx+1}y_i^{my+1}&=\sum x_iy_iz_i\\
&\vdots\\
c_1\sum x_i^{mx}y_i^{my}+c_2\sum x_i^{mx+1}y_i^{my}+\ldots+c_k\sum x_i^{mx}y_i^{my+1}+c_{k+1}\sum x_i^{mx+1}y_i^{my+1}+\ldots&\\
+c_{(mx+1)(my+1)}\sum x_i^{2mx}y_i^{2my}&=\sum x_i^{mx}y_i^{my}z_i
\end{aligned} \tag{48}$$

The *normal equations* represented in matrix form are:

$$
\begin{bmatrix}
\sum x_i^0 y_i^0 & \sum x_i^1 y_i^0 & \cdots & \sum x_i^0 y_i^1 & \sum x_i^1 y_i^1 & \cdots & \sum x_i^{mx} y_i^{my} \\
\sum x_i^1 y_i^0 & \sum x_i^2 y_i^0 & \cdots & \sum x_i^1 y_i^1 & \sum x_i^2 y_i^1 & \cdots & \sum x_i^{mx+1} y_i^{my} \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
\sum x_i^0 y_i^1 & \sum x_i^1 y_i^1 & \cdots & \sum x_i^0 y_i^2 & \sum x_i^1 y_i^2 & \cdots & \sum x_i^{mx} y_i^{my+1} \\
\sum x_i^1 y_i^1 & \sum x_i^2 y_i^1 & \cdots & \sum x_i^1 y_i^2 & \sum x_i^2 y_i^2 & \cdots & \sum x_i^{mx+1} y_i^{my+1} \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
\sum x_i^{mx} y_i^{my} & \sum x_i^{mx+1} y_i^{my} & \cdots & \sum x_i^{mx} y_i^{my+1} & \sum x_i^{mx+1} y_i^{my+1} & \cdots & \sum x_i^{2mx} y_i^{2my}
\end{bmatrix}
\bullet
\begin{bmatrix}
c_1 \\ c_2 \\ \vdots \\ c_k \\ c_{k+1} \\ \vdots \\ c_{(mx+1)(my+1)}
\end{bmatrix}
=
\begin{bmatrix}
\sum x_i^0 y_i^0 z_i \\ \sum x_i^1 y_i^0 z_i \\ \vdots \\ \sum x_i^0 y_i^1 z_i \\ \sum x_i^1 y_i^1 z_i \\ \vdots \\ \sum x_i^{mx} y_i^{my} z_i
\end{bmatrix}
\tag{49}
$$

Here we have a $(m_x+1)(m_y+1) \times (m_x+1)(m_y+1)$ matrix of known quantities multiplying an unknown $(m_x+1)(m_y+1) \times 1$ coefficient vector to obtain a known $(m_x+1)(m_y+1) \times 1$ right-hand-side vector. To build the above set of equations within a computer code, one begins by forming the $n \times (m_x+1)(m_y+1)$ matrix **C** and its $(m_x+1)(m_y+1) \times n$ transpose $\mathbf{C}^\mathrm{T}$:

$$
\mathbf{C} =
\begin{bmatrix}
x_1^0 y_1^0 & x_1^1 y_1^0 & \cdots & x_1^0 y_1^1 & x_1^1 y_1^1 & \cdots & x_1^{mx} y_1^{my} \\
x_2^0 y_2^0 & x_2^1 y_2^0 & \cdots & x_2^0 y_2^1 & x_2^1 y_2^1 & \cdots & x_2^{mx} y_2^{my} \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
x_n^0 y_n^0 & x_n^1 y_n^0 & \cdots & x_n^0 y_n^1 & x_n^1 y_n^1 & \cdots & x_n^{mx} y_n^{my}
\end{bmatrix}
\quad \text{and}
$$

$$
\mathbf{C}^\mathrm{T} =
\begin{bmatrix}
x_1^0 y_1^0 & x_2^0 y_2^0 & \cdots & x_n^0 y_n^0 \\
x_1^1 y_1^0 & x_2^1 y_2^0 & \cdots & x_n^1 y_n^0 \\
\vdots & \vdots & \vdots & \vdots \\
x_1^0 y_1^1 & x_2^0 y_2^1 & \cdots & x_n^0 y_n^1 \\
x_1^1 y_1^1 & x_2^1 y_2^1 & \cdots & x_n^1 y_n^1 \\
\vdots & \vdots & \vdots & \vdots \\
x_1^{mx} y_1^{my} & x_2^{mx} y_2^{my} & \cdots & x_n^{mx} y_n^{my}
\end{bmatrix}
. \tag{50}
$$

Then, the *normal equations* defined in Eqs. 49 can be easily developed using matrix multiplication as follows:

$$\mathbf{C}^{\mathrm{T}}\mathbf{C}\ \mathbf{c} = \mathbf{C}^{\mathrm{T}}\mathbf{z}\ , \tag{51}$$

and the known matrices $\mathbf{C}^{\mathrm{T}}\mathbf{C}$ and $\mathbf{C}^{\mathrm{T}}\mathbf{z}$ can be passed to a simultaneous equations solver to solve for the unknown coefficient vector **c**. Note that when forming the **C** matrix that columns are formed by varying the first superscript from 0 to m_x, then repeating this pattern incrementing values of the second superscript from 0 to m_y. If one desires to omit one or more terms from the fit equation, the corresponding columns of the **C** matrix and the corresponding rows of the $\mathbf{C}^{\mathrm{T}}$ matrix are omitted prior to forming the $\mathbf{C}^{\mathrm{T}}\mathbf{C}$ and $\mathbf{C}^{\mathrm{T}}\mathbf{z}$ matrices and the number of coefficients solved for is reduced accordingly.

The quantities which describe the *goodness* of the fit in Eqs 8 and 9 are for this case:

$$\begin{aligned}
\sigma_{z,xy} &= \left[\frac{1}{n-(m_x+1)(m_y+1)}\sum_{i=1}^{n}\left(z_i-\hat{z}_i\right)^2\right]^{1/2} \\
\sigma_{y,x} &= \left[\frac{1}{n-(m_x+1)(m_y+1)}\sum_{i=1}^{n}\left(z_i-c_1-c_2x_i-\ldots-c_ky_i-c_{k+1}x_iy_i-\ldots-c_{(mx+1)(my+1)}x_i^{mx}y_i^{my}\right)^2\right]^{1/2}
\end{aligned} \tag{52}$$

and the correlation coefficient, R, which is computed from

$$R = \left[1-\frac{\sigma_{z,xy}^2}{\sigma_z^2}\right]^{1/2} \quad \text{where the standard deviation of } z, \quad \sigma_z = \left[\frac{1}{n-1}\sum_{i=1}^{n}\left(z_i-\bar{z}\right)^2\right]^{1/2} . \tag{53}$$

Uncertainty in Surface Fit Coefficients

The application of the uncertainty method requires n triplets of experimental data $\left(x_i, y_i, z_i\right)$, along with associated bias errors, $\left(B_x\right)_i$, $\left(B_y\right)_i$ and $\left(B_z\right)_i$, and precision errors, $\left(P_x\right)_i$, $\left(P_y\right)_i$ and $\left(P_z\right)_i$. The bias uncertainty propagated into the k^{th} coefficient c_k, denoted by $\left(B_c\right)_k$, is computed from:

$$\begin{aligned}
\left(B_c\right)_k^2 &= \sum_{i=1}^{n}\left(\frac{\partial c_k}{\partial x_i}\right)^2\left(B_x\right)_i^2 + \sum_{i=1}^{n}\left(\frac{\partial c_k}{\partial y_i}\right)^2\left(B_y\right)_i^2 + \sum_{i=1}^{n}\left(\frac{\partial c_k}{\partial z_i}\right)^2\left(B_z\right)_i^2 \\
&+2\sum_{i=1}^{n-1}\sum_{j=i+1}^{n}\left(\frac{\partial c_k}{\partial x_i}\right)\left(\frac{\partial c_k}{\partial x_j}\right)\left(B_x\right)_i\left(B_x\right)_j + 2\sum_{i=1}^{n-1}\sum_{j=i+1}^{n}\left(\frac{\partial c_k}{\partial x_i}\right)\left(\frac{\partial c_k}{\partial y_j}\right)\left(B_x\right)_i\left(B_y\right)_j \\
&+2\sum_{i=1}^{n-1}\sum_{j=i+1}^{n}\left(\frac{\partial c_k}{\partial y_i}\right)\left(\frac{\partial c_k}{\partial y_j}\right)\left(B_y\right)_i\left(B_y\right)_j + 2\sum_{i=1}^{n-1}\sum_{j=i+1}^{n}\left(\frac{\partial c_k}{\partial y_i}\right)\left(\frac{\partial c_k}{\partial z_j}\right)\left(B_y\right)_i\left(B_z\right)_j \\
&+2\sum_{i=1}^{n-1}\sum_{j=i+1}^{n}\left(\frac{\partial c_k}{\partial z_i}\right)\left(\frac{\partial c_k}{\partial z_j}\right)\left(B_z\right)_i\left(B_z\right)_j + 2\sum_{i=1}^{n-1}\sum_{j=i+1}^{n}\left(\frac{\partial c_k}{\partial z_i}\right)\left(\frac{\partial c_k}{\partial x_j}\right)\left(B_z\right)_i\left(B_x\right)_j
\end{aligned} , \tag{54}$$

where we have assumed correlated biases among all combinations of x_i, y_i and z_i. If any of these biases among the raw data values can be shown to be uncorrelated, then the corresponding terms in Eq. 54 should be excluded.

The precision uncertainty propagated into the coefficient c_k, $\left(P_c\right)_k$, is obtained from a similar equation; however, since precision errors are considered to be random, all precision errors among the raw data values are uncorrelated.

$$\left(P_c\right)_k^2 = \sum_{i=1}^{n}\left(\frac{\partial c_k}{\partial x_i}\right)^2\left(P_x\right)_i^2 + \sum_{i=1}^{n}\left(\frac{\partial c_k}{\partial y_i}\right)^2\left(P_y\right)_i^2 + \sum_{i=1}^{n}\left(\frac{\partial c_k}{\partial z_i}\right)^2\left(P_z\right)_i^2 . \tag{55}$$

The total uncertainty, $\left(U_c\right)_k$, for the coefficient c_k is then computed using either of the formulas given in Eqs. 14 or 15.

Uncertainty in Predicted Values from Surface Fit

After calculating the coefficients of the surface fit equation, one can supply new values X and Y in order to compute a new value Z using:

$$Z(X,Y) = c_1 + c_2 X + \ldots + c_k Y + c_{k+1} XY + \ldots + c_{(mx+1)(my+1)} X^{mx} Y^{my}$$
$$\text{for } X \text{ and } Y \text{ chosen such that} \quad x_{\min} \le X \le x_{\max} \quad \text{and} \quad y_{\min} \le Y \le y_{\max} \tag{56}$$

The most general equation to define the bias uncertainty that will propagate into this fitted $Z(X,Y)$ is found to be:

$$\begin{aligned}
B_Z^2 = &\sum_{i=1}^{n}\left(\frac{\partial Z}{\partial x_i}\right)^2 (B_x)_i^2 + \sum_{i=1}^{n}\left(\frac{\partial Z}{\partial y_i}\right)^2 (B_y)_i^2 + \sum_{i=1}^{n}\left(\frac{\partial Z}{\partial z_i}\right)^2 (B_z)_i^2 \\
&+2\sum_{i=1}^{n-1}\sum_{j=i+1}^{n}\left(\frac{\partial Z}{\partial x_i}\right)\left(\frac{\partial Z}{\partial x_j}\right)(B_x)_i (B_x)_j + 2\sum_{i=1}^{n-1}\sum_{j=i+1}^{n}\left(\frac{\partial Z}{\partial x_i}\right)\left(\frac{\partial Z}{\partial y_j}\right)(B_x)_i (B_y)_j \\
&+2\sum_{i=1}^{n-1}\sum_{j=i+1}^{n}\left(\frac{\partial Z}{\partial y_i}\right)\left(\frac{\partial Z}{\partial y_j}\right)(B_y)_i (B_y)_j + 2\sum_{i=1}^{n-1}\sum_{j=i+1}^{n}\left(\frac{\partial Z}{\partial y_i}\right)\left(\frac{\partial Z}{\partial z_j}\right)(B_y)_i (B_z)_j \\
&+2\sum_{i=1}^{n-1}\sum_{j=i+1}^{n}\left(\frac{\partial Z}{\partial z_i}\right)\left(\frac{\partial Z}{\partial z_j}\right)(B_z)_i (B_z)_j + 2\sum_{i=1}^{n-1}\sum_{j=i+1}^{n}\left(\frac{\partial Z}{\partial z_i}\right)\left(\frac{\partial Z}{\partial x_j}\right)(B_z)_i (B_x)_j \\
&+\left(\frac{\partial Z}{\partial X}\right)^2 B_X^2 + \left(\frac{\partial Z}{\partial Y}\right)^2 B_Y^2 \\
&+2\sum_{i=1}^{n}\left(\frac{\partial Z}{\partial X}\right)\left(\frac{\partial Z}{\partial x_i}\right) B_X (B_x)_i + 2\sum_{i=1}^{n}\left(\frac{\partial Z}{\partial X}\right)\left(\frac{\partial Z}{\partial y_i}\right) B_X (B_y)_i \\
&+2\sum_{i=1}^{n}\left(\frac{\partial Z}{\partial X}\right)\left(\frac{\partial Z}{\partial z_i}\right) B_X (B_z)_i + 2\sum_{i=1}^{n}\left(\frac{\partial Z}{\partial Y}\right)\left(\frac{\partial Z}{\partial x_i}\right) B_Y (B_x)_i \\
&+2\sum_{i=1}^{n}\left(\frac{\partial Z}{\partial Y}\right)\left(\frac{\partial Z}{\partial y_i}\right) B_Y (B_y)_i + 2\sum_{i=1}^{n}\left(\frac{\partial Z}{\partial Y}\right)\left(\frac{\partial Z}{\partial z_i}\right) B_Y (B_z)_i
\end{aligned} \tag{57}$$

where we have again assumed the most general case of all combinations of correlated biases between X and Y, and the x_i, y_i and z_i. If any of these biases can be shown to be uncorrelated then the appropriate terms in Eq. 57 should be excluded.

The precision uncertainty that propagates into the fitted $Z(X,Y)$ is obtained from

$$
\begin{aligned}
P_Z^2 &= \sum_{i=1}^{n}\left(\frac{\partial Z}{\partial x_i}\right)^2 (P_x)_i^2 + \sum_{i=1}^{n}\left(\frac{\partial Z}{\partial y_i}\right)^2 (P_y)_i^2 + \sum_{i=1}^{n}\left(\frac{\partial Z}{\partial z_i}\right)^2 (P_z)_i^2 \\
&+ \left(\frac{\partial Z}{\partial X}\right)^2 P_X^2 + \left(\frac{\partial Z}{\partial Y}\right)^2 P_Y^2
\end{aligned}
\tag{58}
$$

To evaluate these expressions, we need to determine the partial derivatives: $\frac{\partial Z}{\partial x_i}$, $\frac{\partial Z}{\partial y_i}$ and $\frac{\partial Z}{\partial z_i}$ along with $\frac{\partial Z}{\partial X}$ and $\frac{\partial Z}{\partial Y}$. The derivatives are applied to Eq. 56 and are found to be:

$$
\begin{aligned}
\frac{\partial Z}{\partial x_i} &= \frac{\partial c_1}{\partial x_i} + \frac{\partial c_2}{\partial x_i} X + \ldots + \frac{\partial c_k}{\partial x_i} Y + \frac{\partial c_{k+1}}{\partial x_i} XY + \ldots + \frac{\partial c_{(mx+1)(my+1)}}{\partial x_i} X^{mx} Y^{my} \\
&= \sum_{p=0}^{mx} \sum_{q=0}^{my} \frac{\partial c_k}{\partial x_i} X^p Y^q \text{ , where } k = q\left(m_x + 1\right) + p + 1
\end{aligned}
\quad , \tag{59}
$$

with similar expressions for $\frac{\partial Z}{\partial y_i}$ and $\frac{\partial Z}{\partial z_i}$. For $\frac{\partial Z}{\partial X}$ and $\frac{\partial Z}{\partial Y}$, we have:

$$
\frac{\partial Z}{\partial X} = \sum_{p=1}^{mx} \sum_{q=0}^{my} p\, c_k\, X^{p-1}\, Y^q \text{ and } \frac{\partial Z}{\partial Y} = \sum_{p=0}^{mx} \sum_{q=1}^{my} q\, c_k\, X^p\, Y^{q-1}, \text{ where } k = q\left(m_x + 1\right) + p + 1 \tag{60}
$$

Note that $\frac{\partial Y}{\partial x_i}$, $\frac{\partial Y}{\partial y_i}$ and $\frac{\partial Y}{\partial X}$ are functions of X and Y and must be recomputed for each new value of X or Y.

Computation of the expressions in Eqs. 54-55 and 57-59 require the sensitivity derivatives: $\frac{\partial c_k}{\partial x_i}$ $\frac{\partial c_k}{\partial y_i}$ and $\frac{\partial c_k}{\partial z_i}$. Using the method described earlier, we form analytic expressions for the coefficients using Cramer's rule which makes use of the $\mathbf{C}^{\mathrm{T}}\mathbf{C}$ and $\mathbf{C}^{\mathrm{T}}\mathbf{z}$ matrices which are described in Eqs. 49-51. The expressions for the coefficients will be similar in form to Eqs. 37. One computes derivatives of these quantities in a manner analogous to that described in Eqs. 40-43. The only change is in the implementation details of steps 3 and 7 of the procedure previously outlined; namely, the replacement of the contents of a column in a matrix with its derivative. This is a minor change; however, if one further implements the ability to pick and choose any terms in the desired functional form of the surface fit, then one must keep track of the omitted terms when incorporating this portion of the algorithm. In all other respects the calculation of the sensitivity derivatives is straightforward and identical to that discussed previously.

Now, given the values X and Y, we can compute a value Z from Eq. 56 and an associated B_Z and P_Z from Eqs. 57 and 58. From the latter two quantities we can determine a total uncertainty U_Z from

Eqs. 14 or 15. This process should be repeated for X_j and Y_j which vary throughout the domain $x_{\min} \le X_j \le x_{\max}$ and $y_{\min} \le Y_j \le y_{\max}$ and a series of $Z(X,Y)_j$ and $\left(U_Z\right)_j$ will result. Then, the uncertainty in fitted $Z(X,Y)_j$ values can be plotted along with the fitted surface as follows. Plot the original $\left(x_i, y_i, z_i\right)$ data values along with the fitted surface $Z(X,Y)_j$ obtained from the new $\left(X_j, Y_j\right)$ data pairs. Then plot above and below this fitted surface the two additional surfaces: $Z(X,Y)_j + \left(U_Z\right)_j$ and $Z(X,Y)_j - \left(U_Z\right)_j$. The latter two surfaces then give an indication of the total uncertainty in fitted $Z(X,Y)_j$ values across the entire domain $x_{\min} \le X_j \le x_{\max}$ and $y_{\min} \le Y_j \le y_{\max}$. Since a global plot of this sort may prove difficult to extract quantitative information from, one may wish to alternatively plot two-dimensional slices of the data to provide local uncertainty behavior.

Examples

This section gives a few simple examples of the uncertainty that one may expect in coefficients and fitted values using the procedures described in the previous sections. The results provide some insight and quantification of the penalties that can arise as a result of improper use of regression techniques. The approach taken is to use a basic set of experimental data with prescribed uncertainty and to perform a fit to the data and compute the uncertainty propagated through the fit equations. Then, the data set is altered in a variety of ways to determine the resulting changes in the uncertainty results. The data to be fitted are given in Table 1 and include only precision uncertainty in the ordinate. The data are taken from pp. 183-184 of the text by Coleman and Steele[2] and are used with permission.

Table 1. Test data for examples.

x_i	$\left(B_x\right)_i$	$\left(P_x\right)_i$	y_i	$\left(B_y\right)_i$	$\left(P_y\right)_i$
2.0	0.0	0.0	2.4	0.0	1.0
3.0	0.0	0.0	3.0	0.0	1.0
4.5	0.0	0.0	3.5	0.0	1.0
5.3	0.0	0.0	4.5	0.0	1.0
6.5	0.0	0.0	4.9	0.0	1.0
7.8	0.0	0.0	5.6	0.0	1.0
8.5	0.0	0.0	6.8	0.0	1.0
10.1	0.0	0.0	7.3	0.0	1.0

The first case consists of a linear fit to the data in Table 1. Following the summary provided in a previous section, the fit coefficients and the total uncertainties (using Eq. 14) in the fit coefficients were computed. The latter quantities were divided by the respective values of the fit coefficients to produce percentage uncertainties, and these data may be found in Table 2. For the given number of data points

and for the specified level of uncertainty in the ordinates, the uncertainty in the intercept is 86% and in the slope, 22%.

Table 2. Uncertainty in fit coefficients.

	Type	c_1	c_2	c_3	Uc_1/c_1 (%)	Uc_2/c_2 (%)	Uc_3/c_3 (%)
Case 1	Linear	1.0278	0.6243	n/a	85.93	21.74	n/a
Case 2	Quadratic	1.3634	0.4857	0.0116	137.68	143.51	493.46
Case 3	Dble Data	0.9855	0.6309	n/a	69.01	16.74	n/a
Case 4	Half Unc	1.0278	0.6243	n/a	42.97	10.87	n/a
Case 5	Add Err	1.0278	0.6243	n/a	88.20	22.13	n/a

The next step is to produce a series a fitted values, $Y(X)_j$, using a series of X_j lying between the minimum and maximum values of the data in the first column of Table 1. Twenty such values were chosen, and since the original data values contained no uncertainty in the abscissa, the uncertainty associated with new abscissas was estimated to be zero. Then, for each of these fitted values, the total uncertainty, $(U_Y)_j$, was calculated and the quantities $Y(X)_j+(U_Y)_j$ and $Y(X)_j-(U_Y)_j$ were formed.

Table 3. Uncertainty in fitted values.

X_N	Y_N	UY_N	Y_N+UY_N	Y_N-UY_N
2.00	2.28	0.64	2.92	1.63
2.43	2.54	0.60	3.14	1.95
2.85	2.81	0.55	3.36	2.26
3.28	3.07	0.51	3.58	2.57
3.71	3.34	0.47	3.81	2.87
4.13	3.61	0.43	4.04	3.17
4.56	3.87	0.40	4.27	3.47
4.98	4.14	0.38	4.52	3.76
5.41	4.41	0.36	4.77	4.04
5.84	4.67	0.35	5.03	4.32
6.26	4.94	0.36	5.29	4.58
6.69	5.20	0.37	5.57	4.84
7.12	5.47	0.39	5.86	5.08
7.54	5.74	0.41	6.15	5.32
7.97	6.00	0.45	6.45	5.56
8.39	6.27	0.48	6.75	5.78
8.82	6.53	0.52	7.06	6.01
9.25	6.80	0.57	7.37	6.23
9.67	7.07	0.62	7.68	6.45
10.10	7.33	0.66	8.00	6.67

This data may be found in Table 3. One can see that the uncertainty propagated into fitted values varies with the abscissa from a maximum of 0.66 near each end of the range to a minimum of 0.35 approximately midway through the range. Figure 1 shows a plot of this behavior. Also, contained in Fig. 1 is the original data with error bars representing the precision uncertainty in the ordinates along with the linear fit satisfying the least squares criterion.

In addition, quantities which describe the *goodness* of the fit: the standard error (also known as the standard error of the estimate, SEE) and the correlation coefficient, were computed from Eqs. 8 and 9, respectively. The results may be found in Table 4. The total uncertainty in the abscissa for fitted values, estimated to be zero, is listed in the table as well as an average uncertainty for the ordinate. Finally, for comparison, the last column of Table 4 contains the standard error multiplied by two; this number has often been used in the past as the uncertainty bands around fitted values from the fit.

Table 4. Goodness of fit parameters.

	Type	R	Std Err	UX_N	Avg UY_N	2 SEE
Case 1	Linear	0.9863	0.2905	0	0.4759	0.5810
Case 2	Quadratic	0.9848	0.3051	0	0.5653	0.6101
Case 3	Dble Data	0.9903	0.2234	0	0.3574	0.4468
Case 4	Half Unc	0.9863	0.2905	0	0.2380	0.5810
Case 5	Add Err	0.9863	0.2905	0.3162	0.5303	0.5810

A glance at Fig. 1 shows that the fit appears to be reasonably good, and the fit parameters bear this out with a high correlation coefficient and a relatively small standard error of the original data points about the fitted line. What these statistics do **not** indicate are the high uncertainties in the fit coefficients, which are presumed to result from: the amount of data used to construct the fit and the high uncertainties associated with the ordinates of the original data. Fitted values obtained from this fit will have a total uncertainty associated with them which varies from 0.66 to 0.35 or an average value of 0.48. This latter value is greater than the standard error, but less than the uncertainties in the ordinates of the original data values and not as conservative as the less accurate 2 SEE measure applied in the past.

Although the data appear to be reasonably approximated by a linear fit, one might wish to improve the fit by using a higher order polynomial. One might argue that the coefficient multiplying the quadratic term, if not needed, will simply turn out to be a small number, and the higher order fit will do no harm. For case 2, a quadratic fit was computed for the data in Table 1 using the same procedure as described above. The coefficients and percent uncertainty for the coefficients are given in Table 2, the fit parameters are in Table 4, and a plot of the results may be found in Fig. 2. We see that for this particular data set, the standard error and the correlation coefficient are nearly the same for the quadratic and linear cases. However, the uncertainties associated with the coefficients, particularly the quadratic term, are *very high.* The uncertainty in fitted values varies from a maximum of 0.88 to a minimum of 0.47 which yields an average value of 0.57, and this is higher than that for a linear fit with no improvement in fit characteristics.

In an attempt to reduce uncertainty, the input data set listed in Table 1 was approximately doubled in size, and a linear fit was computed for this extended data set for case 3. The extended data set was obtained by adding an additional pair of data values between each of the existing data values by means of linear interpolation in each coordinate, thereby increasing the data set from 8 to 15 data pairs. Each new data pair was assigned zero bias and precision uncertainties for the abscissas and a zero bias and a precision uncertainty equal to one for each ordinate. One might argue that the manner in which the new data were generated may bias the results, nevertheless, the improvement was surprisingly meager. The coefficients and percent uncertainty for the coefficients are given in Table 2, and the fit parameters are in Table 4. The computed coefficients varied slightly due to the change in the input data, but were close enough to case 1 to afford a reasonable comparison. There was some improvement in the uncertainty in the fit coefficients: a reduction from 86% to 69% for the intercept and a reduction from 22% to 17% for the slope. Similarly, the uncertainty in fitted values varied from a maximum of 0.51 to a minimum of 0.26 for an average value of 0.36. The correlation coefficient improved from 0.986 to 0.99 and the standard error decreased from 0.29 to 0.22.

The fourth example was designed to test the effect of halving the uncertainty in the input data. Thus, the eight data pairs listed in Table 1 were used with the exception that the precision uncertainties for the ordinates were reduced to 0.5. The results of the linear fit are listed in the appropriate tables, and we find a marked decrease for this case when compared to case 1. The uncertainty in the intercept was reduced from 86% to 43% and for the slope from 22% to 11%. The uncertainty in fitted values varied from a maximum of 0.33 to a minimum of 0.18 for an average value of 0.24. The correlation coefficient and standard error were identical to that of the first case because these parameters are formed from the input coordinate pairs which did not change. Notice the considerable difference between the average uncertainty in the ordinate, 0.24, and the 2 SEE parameter which remained the same as for case 1 at 0.58; the 2 SEE parameter is an excessively conservative bound for this case. For a given fit order, a reduction in uncertainty in the data to be fitted is far more effective in reducing uncertainty associated with fit coefficients and with fitted values than simply increasing the amount of data to be fitted.

Table 5. Test data for case 5.

x_i	$(B_x)_i$	$(P_x)_i$	y_i	$(B_y)_i$	$(P_y)_i$
2.0	0.1	0.3	2.4	0.1	1.0
3.0	0.1	0.3	3.0	0.1	1.0
4.5	0.1	0.3	3.5	0.1	1.0
5.3	0.1	0.3	4.5	0.1	1.0
6.5	0.1	0.3	4.9	0.1	1.0
7.8	0.1	0.3	5.6	0.1	1.0
8.5	0.1	0.3	6.8	0.1	1.0
10.1	0.1	0.3	7.3	0.1	1.0

Finally, for the fifth case, reasonable bias and precision errors were chosen for both the abscissas and the ordinates for the data listed in Table 1 in order to better simulate the case of experimentally derived data.

Although, in general, these errors may vary from data pair to data pair in the set, for simplicity, the same estimates were used for each data pair. The data set used for this example may be found in Table 5.

In order to calculate the bias error propagated into the coefficients $\left(B_c\right)_k$ from Eq. 12, one must consider which biases, if any, are correlated. Since the same numerical values were prescribed for each of the abscissas and ordinates, a reasonable assumption is that the $\left(B_x\right)_i$ are correlated with each other and the $\left(B_y\right)_i$ are correlated with each other, but that the $\left(B_x\right)_i$ are not correlated with the $\left(B_y\right)_i$. For calculating the bias error propagated into fitted values B_Y from Eq. 31, one must estimate the expected bias in the abscissa B_X for a fitted data pair and decide if this bias is correlated with any other biases. B_X was assigned values of 0.1 for each new data pair and was assumed to be correlated with the $\left(B_x\right)_i$ but not with the $\left(B_y\right)_i$. The results are listed in the appropriate tables. The computed fit coefficients are identical to case 1 since the $\left(x_i, y_i\right)$ data pairs are unchanged, and the uncertainty propagated into the fit coefficients is about the same. The correlation coefficient and standard error are unchanged from case 1, and the total uncertainty in the abscissa for fitted values is 0.32 which is computed using Eq. 14. The uncertainty in fitted ordinates varied from a maximum of 0.71 to a minimum of 0.42 for an average value of 0.53 which is about 10% higher than for case 1. The 2 SEE bound remained unchanged. The fact that this case differs little from case 1 indicates that the uncertainty calculations are dominated by the precision uncertainty in the ordinates of the input data and that the largest improvement will result from a decrease in this uncertainty as shown by case 4.

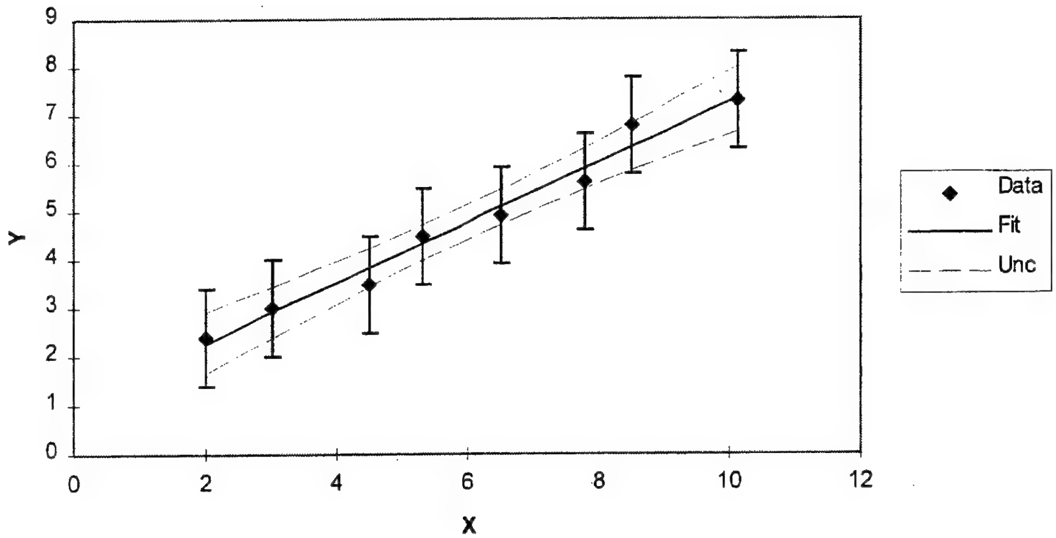

Fig. 1. Linear fit to basic data set.

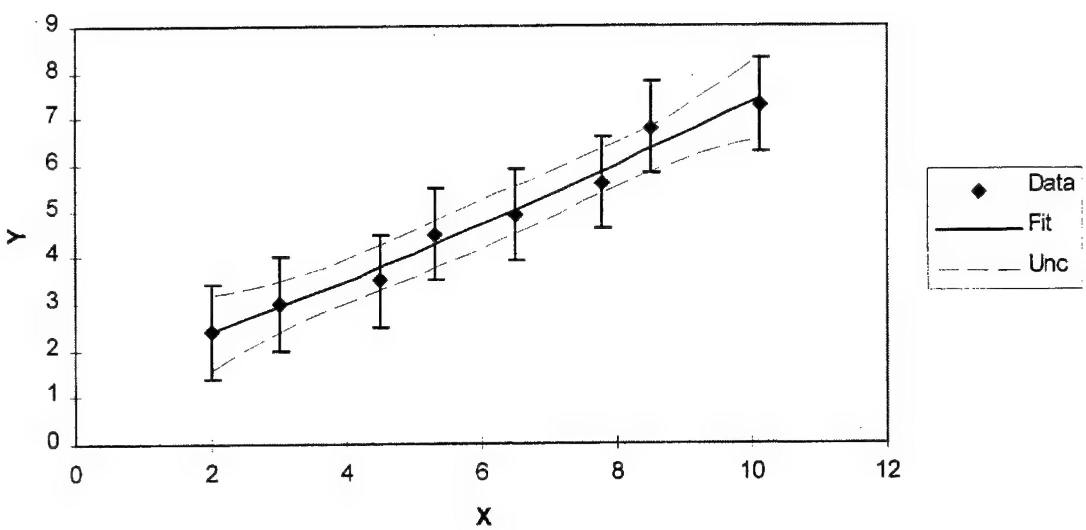

Fig. 2. Quadratic fit to basic data set.

Conclusions

This paper has reviewed the manner in which curve fits and surface fits of arbitrary order which satisfy the least squares criterion are constructed. The method for the computation of uncertainty propagated into fit coefficients and into fitted values obtained from the fit for both curve and surface fits has been outlined. The primary obstacle to the efficient implementation of these calculations: the determination of the sensitivity derivatives, has been removed with the explanation of an analytic method for the calculation of these derivatives for arbitrary order curve and surface fits. A detailed prescription describing one possible approach for the implementation of the method was provided. To illustrate the power of the techniques, simple examples were chosen and contrasted using a generic data set which is typical of experimentally derived data. The results showed the general behavior of the calculations under a variety of conditions. In particular, the danger of fitting a higher order polynomial to a data set when such a model is not warranted, was illustrated. For this particular data set, a reduction in the uncertainty associated with the input data was more effective at producing reductions in uncertainty associated with fit coefficients and fitted values than simply using more data with the original uncertainty levels. Although the equations are somewhat tedious, these methods may be readily implemented within a computer program making the computation of uncertainty for fits employing the method of Least Squares a routine matter.

Acknowledgments

The authors wish to thank Professor W. Glenn Steele for his helpful comments during the course of this work and for the use of data from the text.

Appendix

Testing the Surface Fit Implementation

To test a surface fitting computer code, one commonly generates a set of data triplets, (x_i, y_i, z_i), which satisfy a known function $z = f(x,y)$ formed from the product of two polynomials. Then, the data are input into the program, the user selects m_x and m_y to match the characteristics of the known function, and the output will be a set of coefficients which should match those of the function which generated the data. Specifically, a set of (x_i, y_i) are produced in some convenient manner and then input to the function $z = f(x,y)$ to obtain z_i and complete the data triplets (x_i, y_i, z_i). This seemingly innocuous procedure can lead to frustration because the generation of test data, (x_i, y_i), for input into $z = f(x,y)$ *cannot be chosen arbitrarily* if one desires a unique solution! To see how problems can arise, consider the normal equations, Eqs. 49, for the case $m_x = m_y = 1$ which become:

$$\begin{bmatrix} \sum x_i^0 y_i^0 & \sum x_i^1 y_i^0 & \sum x_i^0 y_i^1 & \sum x_i^1 y_i^1 \\ \sum x_i^1 y_i^0 & \sum x_i^2 y_i^0 & \sum x_i^1 y_i^1 & \sum x_i^2 y_i^1 \\ \sum x_i^0 y_i^1 & \sum x_i^1 y_i^1 & \sum x_i^0 y_i^2 & \sum x_i^1 y_i^2 \\ \sum x_i^1 y_i^1 & \sum x_i^2 y_i^1 & \sum x_i^1 y_i^2 & \sum x_i^2 y_i^2 \end{bmatrix} \begin{bmatrix} c_1 \\ c_2 \\ c_3 \\ c_4 \end{bmatrix} = \begin{bmatrix} \sum x_i^0 y_i^0 z_i \\ \sum x_i^1 y_i^0 z_i \\ \sum x_i^0 y_i^1 z_i \\ \sum x_i^1 y_i^1 z_i \end{bmatrix}. \tag{A1}$$

Now if , for convenience, one were to choose $x_i = y_i$ yielding data triplets of the form (x_i, x_i, z_i), then when developing the coefficient matrix above, row 2 and row 3 would become identical:

$$\begin{matrix} \sum x_i^1 & \sum x_i^2 & \sum x_i^2 & \sum x_i^3 & \text{- row 2} \\ \sum x_i^1 & \sum x_i^2 & \sum x_i^2 & \sum x_i^3 & \text{- row 3} \end{matrix}. \tag{A2}$$

This causes the matrix to become singular with zero determinant. In other words, we have three independent equations in the four unknowns yielding a one-parameter family of solutions instead of a single unique solution. More generally, if one chooses $y_i = ax_i + b$, one can show that $\text{row } 3 = a(\text{row } 2) + b(\text{row } 1)$ which will again lead to a non-unique solution to the system of equations. The point is that the method will fail to produce a unique solution when the x_i and y_i are chosen such that they are linearly dependent. The data to be fitted must satisfy the requirement that $z = f(x,y)$ be a function of two linearly independent variables. This situation is unlikely to occur when fitting experimentally derived data and is usually only encountered when generating artificial data for code validation.

Table A1. Test data generation scheme.

r	s	i	x_i	y_i
1	1	1	1	1
1	2	2	1	2
1	3	3	1	3
2	1	4	2	1
2	2	5	2	2
2	3	6	2	3
3	1	7	3	1
3	2	8	3	2
3	3	9	3	3

One simple scheme for creating test data is to choose an integer, l, such that the desired number of data triplets will be $n = l^2$; then using two loops, with indices r and s, one constructs the x_i and y_i using:

$$\begin{aligned} &x_i = r\,x_0 \text{ and } y_i = s\,y_0 \quad \text{where } i = (r-1)\,l + s \\ &1 \le r \le l,\ 1 \le s \le l,\ 1 \le i \le n;\ x_0 \text{ and } y_0 \text{ are arbitrary scale factors} \end{aligned} \tag{A3}$$

For example, if one chooses $n = 9$ and $x_0 = y_0 = 1.0$, this scheme produces the data shown in Table 1. One then chooses an appropriate $z = f(x,y)$ of the form

$$z = \sum_{p=0}^{mx} \sum_{q=0}^{my} c_k\, x^p y^q \text{, where } k = q\,(m_x + 1) + p + 1 \text{ and } 1 \le k \le (m_x + 1)(m_y + 1) \ , \tag{A4}$$

and specifies m_x and m_y. Finally, the coefficients for this function must be chosen. To avoid large numbers in the coefficient matrix for increasingly large choices of m_x and m_y, one can define the coefficients to be

$$c_k = \frac{(-1)^{k+1}}{k} \ . \tag{A5}$$

Summarizing, one generates n pairs of x_i and y_i using Eq. A3, then inserts these values in Eq. A4 to obtain the z_i to complete each data triplet. When this test data is input into the surface fitting code, it should return the fit coefficients defined in Eq. A5 with appropriate choices for m_x and m_y.

References

1. Coleman, Hugh W. and W. Glenn Steele, Experimentation and Uncertainty Analysis for Engineers, Second edition, John Wiley and Sons, New York (1999).

2. Coleman, Hugh W. and W. Glenn Steele, Experimentation and Uncertainty Analysis for Engineers, John Wiley and Sons, New York (1989).

3. Press, William H., Saul A. Teukolsky, William T. Vetterling and Brian P. Flannery, Numerical Recipes in FORTRAN, Second Edition, Cambridge University Press, New York (1992).

4. Holman, Jack P., Experimental Methods for Engineers, Sixth Edition, McGraw-Hill, New York (1994).

Initial Distribution

Copies		Code	Name
1	NAVSEA		
	1	03H	Fein
2	ONR		
	1	333	Purtell
	1	333	Lekoudis
	1	342	McMullen
2	Academic		
	1		Steele
	1		Coleman
2	DTIC		

Center Distribution

Copies	Code	Name
1	50	Morgan
1	5030	Merrill
1	5060	Walden
1	5200	Stenson
1	5200	Fu
1	5300	Pickett
1	5400	Peterson
1	5500	Motter
1	5600	Koh
1	5600	Ammeen
1	5600	Bedel
1	5600	Cubbage
1	5600	Feldman
1	5600	Faller
1	5600	Junghans
1	5600	Moran
1	5600	Nigon
1	5600	Roddy
1	5600	Russ